**The Mad, Mad World of Matrices**

Last time we discussed:

* scalars versus vectors
* graphical methods
* component representation

**From Vectors to Matrices**

Components of a vector can represent steps in various directions in 2-D or 3-D space.

**A** = ( Ax , Ay ) or **A** = ( Ax , Ay , Az )

Δx , Δy Δx , Δy , Δz

But vectors can be extended beyond components in space to any category of items.

Ex. **Movie tickets sold** = ( 73 , 65 , 82 , 104 , 339 , 452 , 397 )

M , T , W , Th , F , Sa , Su

**In general, a vector is an ordered row of values that follow the rules of vector operations.**

Now imagine extending the vector to another category, so that each value is described by two items.

Ex. M , T , W , Th , F , Sa , Su Week

73 , 65 , 82 , 104 , 339 , 452 , 397 1

**Movie tickets sold** = 58 , 73 , 69 , 82 , 224 , 384 , 251 2

95 , 87 , 77 , 152 , 393 , 556 , 470 3

**An ordered arrangement of values in rows and columns that follow well-defined rules of operation is called a matrix.**

**Example**—Jill sets up a lemonade and cookie stand in her neighborhood. She sells each cup of lemonade for $0.25 and delicious, homemade cookies for $0.50 each. On Saturday she sells 32 cups of lemonade and 20 cookies. On Sunday she sells 24 cups of lemonade and 12 cookies. What are the matrices describing the number of each product sold each day and the price of each product?

Set up the matrices.

|  |  |  |
| --- | --- | --- |
|  | Lemonade | Cookies |
| Saturday |  |  |
| Sunday |  |  |

*Number of Products Sold* =

|  |  |
| --- | --- |
|  | Price |
| Lemonade |  |
| Cookies |  |

*Price of Products* =

**Dimensionality of a Matrix**

A matrix can consist of rows and columns of any length, except that *every row is the same length and so is every column*.

**The dimensionality of a matrix is (#rows)×(#columns)**. (The × is “by,” as in “2 by 4.”)

Ex. A 2×3 matrix (“2 by 3”)

|  |  |  |
| --- | --- | --- |
| 3 | 6 | 4  **2 rows** |
| 7 | -2 | 1 |

**3 columns**

**Exercise:** Identify the dimensionality of the following matrices.

|  |  |
| --- | --- |
| 7 | 3 |
| 26 | 5 |
| 15 | 23 |
| 10 | 17 |
| 11 | 6 |
| 17 | 1 |
| 19 | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 10 | 9 | 8 | 7 | 6 |
| 16 | 17 | 18 | 19 | 20 |
| 15 | 14 | 13 | 12 | 11 |

A matrix where the number of rows equals the number of columns is called a **square matrix** (i.e., 2×2, 3×3, 8×8, etc.).

**Identification of Matrix Entries**

Just as we identify the components of a vector by x, y, z, etc., we have to create a way to identify the individual entries of a matrix.

**Matrix entries are labeled by their row and column, given as Arow column.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A11 | A12 | A13 | A14 | A15 |
| A21 | A22 | A23 | A24 | A25 |
| A31 | A32 | A33 | A34 | A35 |

*A* =

* Entries are counted from left to right and top to bottom, as if reading words on a page.
* The last entry has the same indices as the dimensionality of the matrix.
* The entry Aij is often labeled as the (i, j) entry [e.g., (2, 3) entry]

**Exercise:** Identify the location and value of the circled entries in the matrices on the previous page.

**Matrix Operations**

Matrices have all the same operations as vectors, plus some extras.

**Addition and Subtraction**

Two matrices are added or subtracted by adding or subtracting the corresponding entries in both.

|  |  |
| --- | --- |
| 4 | 7 |
| 7 | 16 |

|  |  |
| --- | --- |
| 1 + 3 | 4 + 3 |
| 2 + 5 | 9 + 7 |

|  |  |
| --- | --- |
| 3 | 3 |
| 5 | 7 |

|  |  |
| --- | --- |
| 1 | 4 |
| 2 | 9 |

+ = = =

Two matrices can be added or subtracted *only if they have the same dimensionality*.

**Multiplication by a Scalar**

When multiplying a matrix by a scalar s, such that *B* = s*A*, multiply each entry by the scalar.

|  |  |
| --- | --- |
| 1s | 4s |
| 2s | 9s |

|  |  |
| --- | --- |
| 1 | 4 |
| 2 | 9 |

*B* = s =

**Matrix Multiplication**

Matrices can be multiplied by each other, but the procedure seems *very strange* at first. Let’s see how it works.

Quick summary—**Each row of the first matrix is multiplied by each column of the second!?**

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |
| 5 | 6 |

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |

**Ex.** *A* = *B* =

|  |
| --- |
| 1 |
| 3 |
| 5 |

The 1st row 1 2 3 is multiplied by the 1st column

and added as follows:

|  |
| --- |
| 1 |
| 6 |
| 15 |

|  |
| --- |
| 1 |
| 3 |
| 5 |

|  |
| --- |
| 1 × |
| 2 × |
| 3 × |

1 2 3

= = 1 + 6 + 15 = 22

Because this result came from the **1st** row and the **1st** column, it becomes the (1, 1) term of the new matrix. Multiplying the **2nd** row by the **1st** column gives the (2, 1) entry, etc. Multiplying all the other rows and columns gives

|  |  |
| --- | --- |
| 22 | 28 |
| 49 | 64 |

*AB* =

Crazy!

**Notes:**

-Not all matrices can be multiplied. The **number of columns of the first matrix must equal the number of rows of the second**.

[2×3] [3×2] → [2×2]

**These two must be the same**.

-The dimensionality of the product is given by the outer numbers.

**Example Redux**—For Jill’s lemonade and cookie stand, how much money did she make each day?

The amount of money made each day is found by multiplying the number sold per product by the price of each product.

|  |  |
| --- | --- |
|  | Price |
| Lemonade |  |
| Cookies |  |

|  |  |  |
| --- | --- | --- |
|  | Lemonade | Cookies |
| Saturday |  |  |
| Sunday |  |  |

*Money Made* = ×

|  |  |
| --- | --- |
|  | Money Made |
| Saturday |  |
| Sunday |  |

=

**Note:** When multiplying two matrices, the categories shared in common disappear in the product.